iPhone OS Networking

Session 107

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

iPhone OS Networking

Joshua Graessley
Senior Software Engineer

What You'll Learn

- Networking API overview
- Major differences from Mac OS X
- Device to device communication
- Mobility techniques
- Performance tips

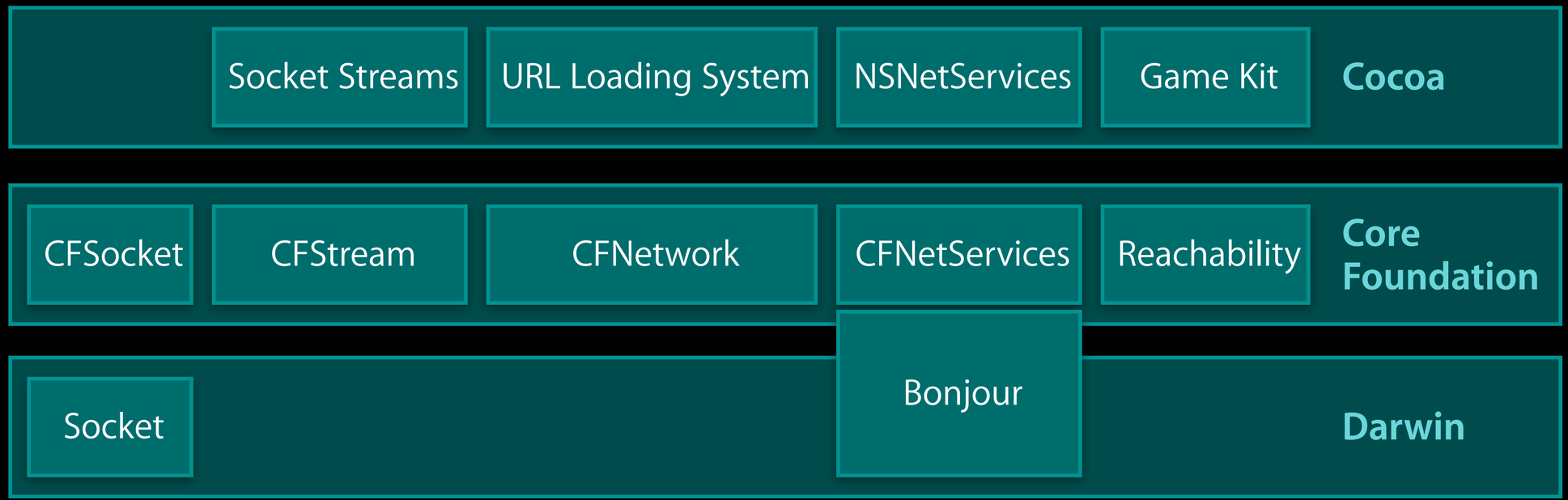
Introduction

- Full networking stack
  - Power
  - Mobility
  - Performance

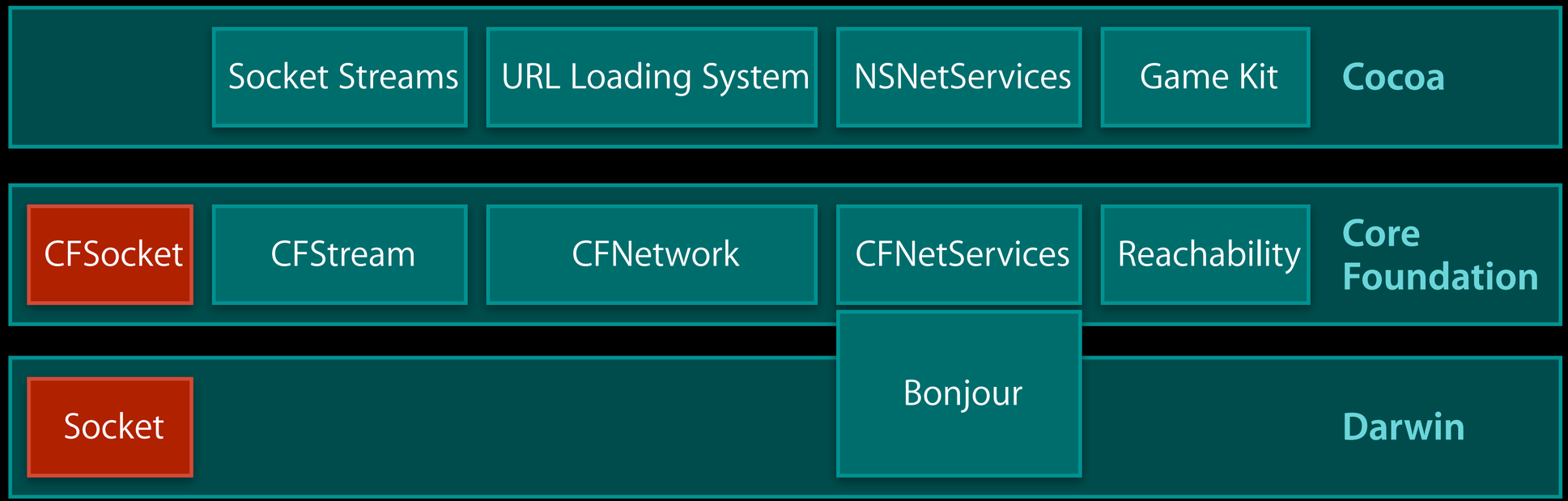
iPhone Environment

- Interfaces
  - WiFi
  - Cellular
    - Except iPods
    - Bluetooth PAN—device to device
    - VPN
- IPv4
- Bonjour

Network APIs



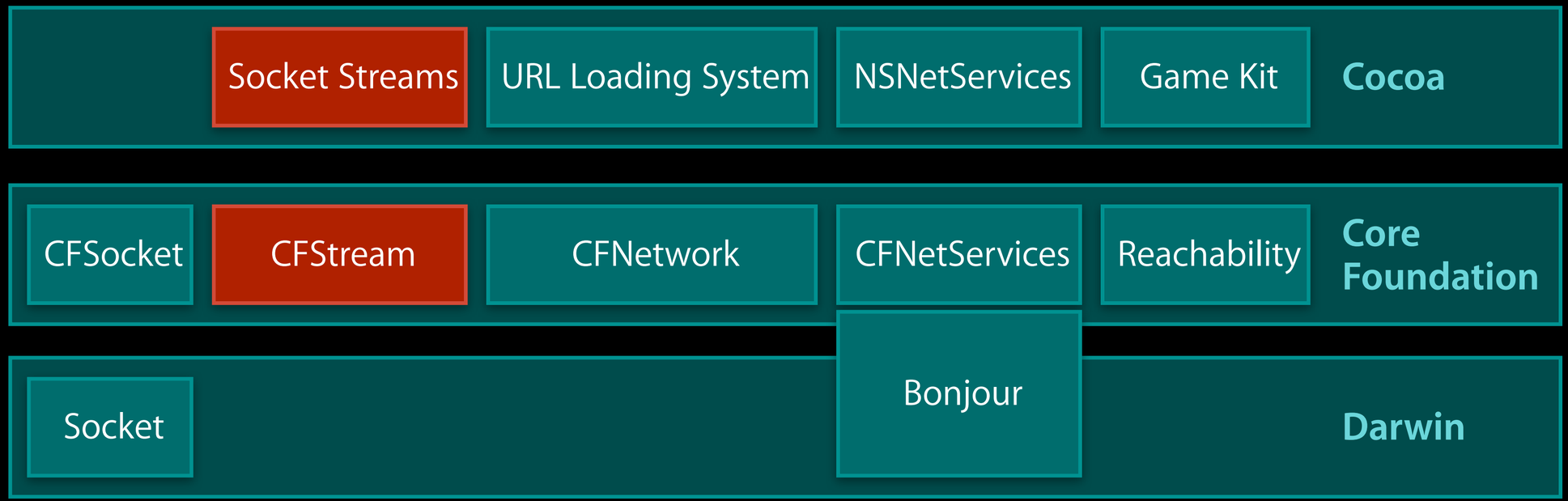
Socket APIs


Socket Streams
URL Loading System
NSNetServices
Game Kit
Cocoa
CFSocket
CFStream
CFNetwork
CFNetServices
Reachability
Core Foundation
Bonjour
Socket
Darwin

Sockets

- Darwin
  - BSD sockets
  - getaddrinfo/getnameinfo, etc.
  - kqueue/select
- Core Foundation
  - Run loop integration
  - CFSocket
  - CFHost

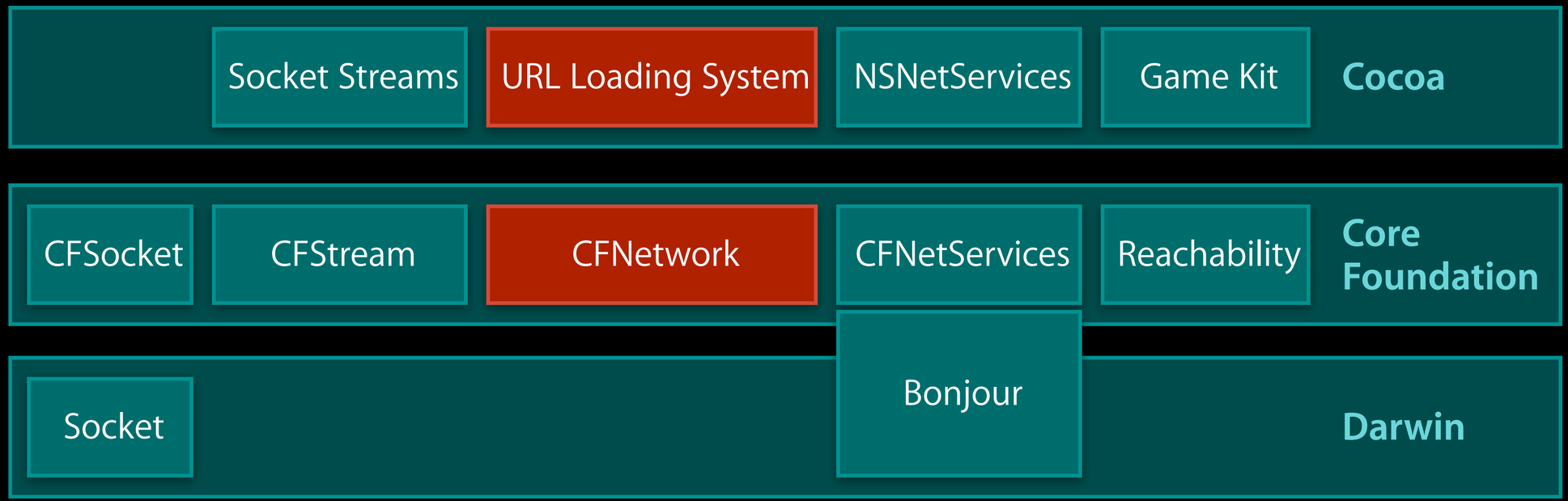
Stream APIs


Socket Streams
URL Loading System
NSNetServices
Game Kit
Cocoa
CFSocket
CFStream
CFNetwork
CFNetServices
Reachability
Core Foundation
Bonjour
Socket
Darwin

Stream Abstraction

- SSL/TLS support
- Proxy support
- Cocoa
  - NSStream
- Core Foundation
  - CFStream

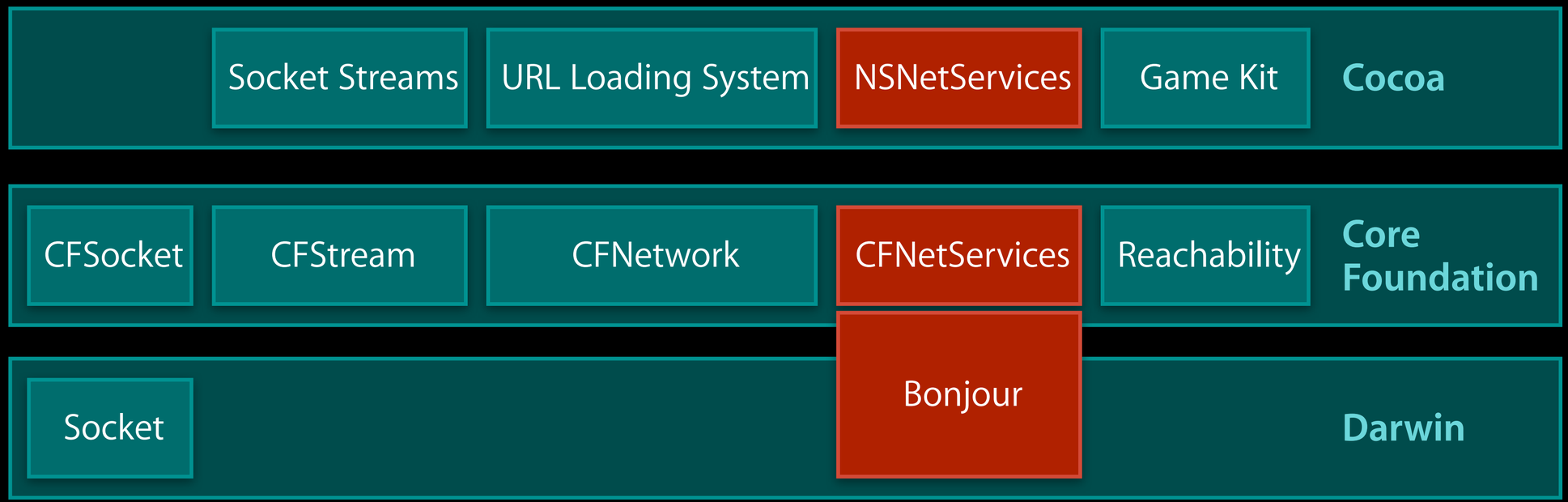
Common Protocol APIs



Common Protocols

- FTP, HTTP, HTTPS
- Cocoa
  - NSURLConnection
  - NSURLResponse
  - NSURLRequest
- Core Foundation
  - CFStream
  - CFHTTPMessage

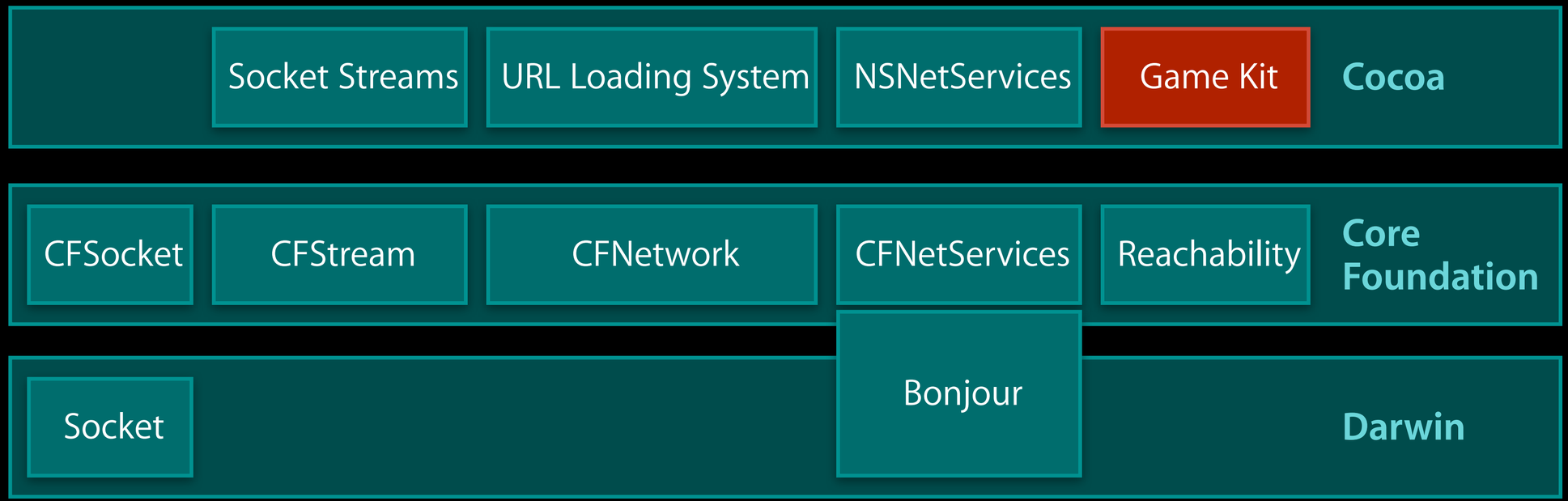
Service Discovery APIs



Service Discovery

- Session 508—Zero Configuration Networking Using Bonjour
- Darwin
  - Bonjour
- Core Foundation
  - Bonjour
  - CFNetService
- Cocoa
  - NSNetService

Miscellaneous APIs



Game Kit

- Session 318—Peer to Peer Networking with Game Kit
- User interface
- Voice chat—Bluetooth and WiFi
- Data sessions—Bluetooth

Unique on iPhone OS

- No root access
  - Read only routing sockets
  - No IPSec policy/association access (PF_KEY)
  - No Berkeley Packet Filter (BPF)
  - Limited Raw IP
  - Low numbered ports
    - No root requirement on iPhone OS 3.0
- Subset of System Configuration Framework
  - Network reachability
  - Use CFProxySupport

New in iPhone OS 3.0

- Captive network support
- Scoped routing
- Bluetooth device to device

Captive Network Support

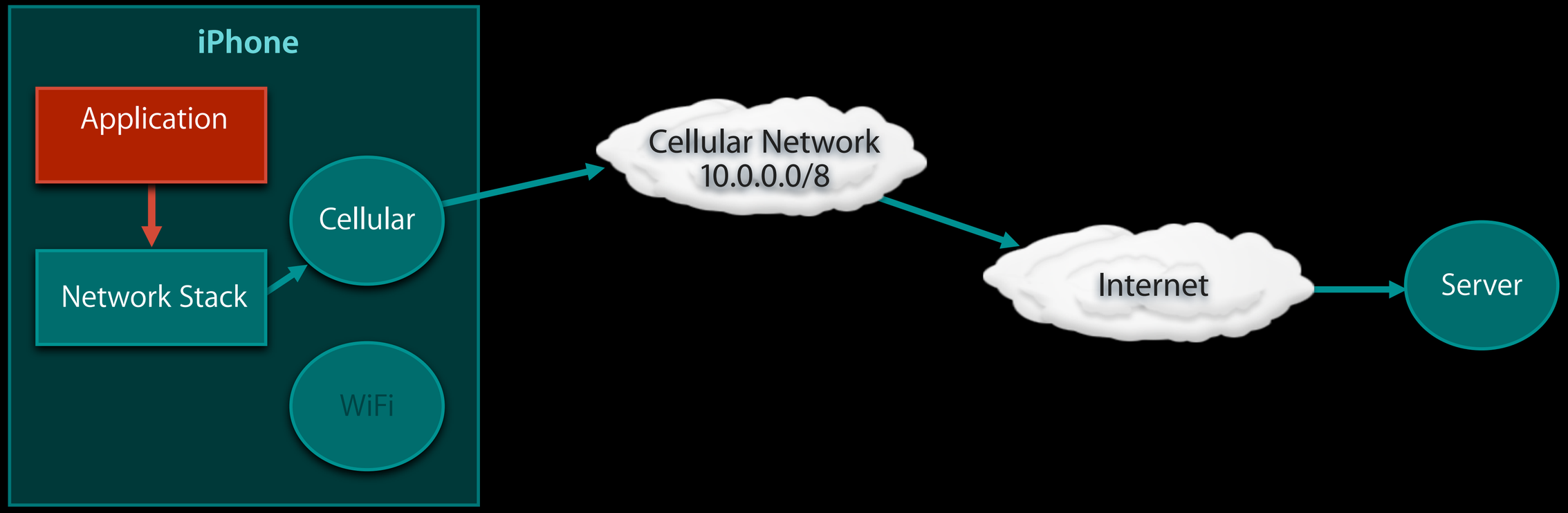
- Avoid broken network state
- Probes network on associate
- Wireless ISP roaming (WISPr) authentication
- Displays web sheet or authenticates
- Scrapes credentials from web sheet
- Disassociate if authentication fails

Scoped Routing

- Source address based routing
- Flexibility—bind to interface
  - Visual voicemail
  - Push notifications
  - Exchange support
  - Internet tethering
  - Game Kit
- New in SnowLeopard

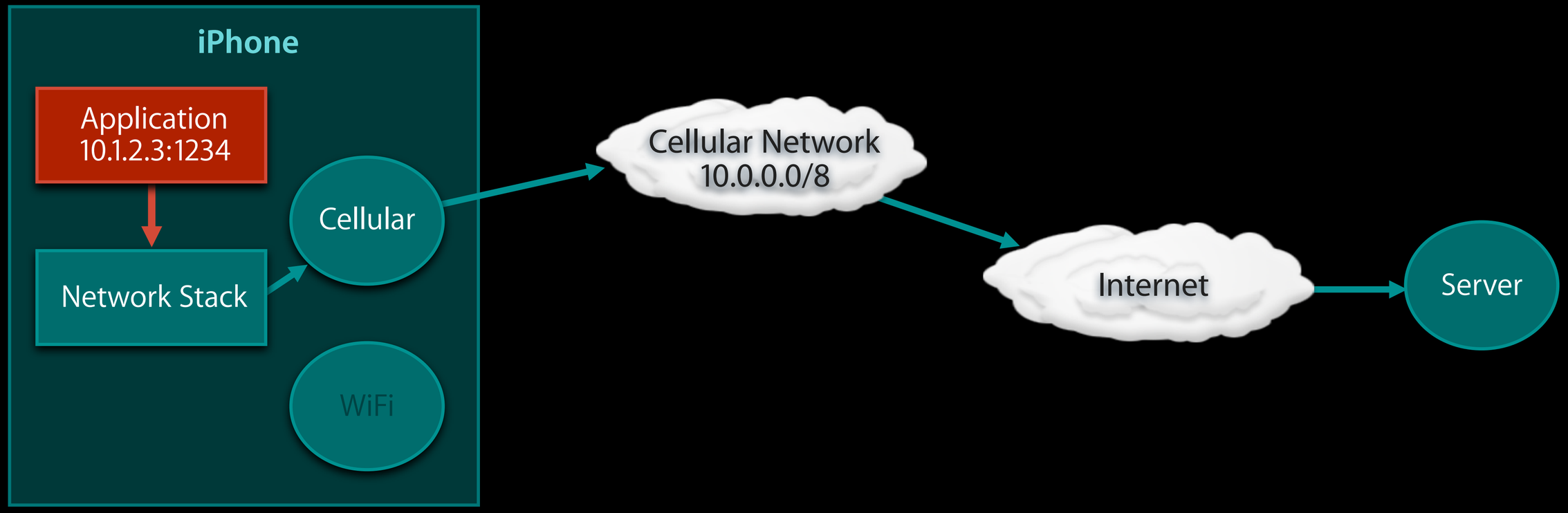
How It Works Today

Application connects over cellular



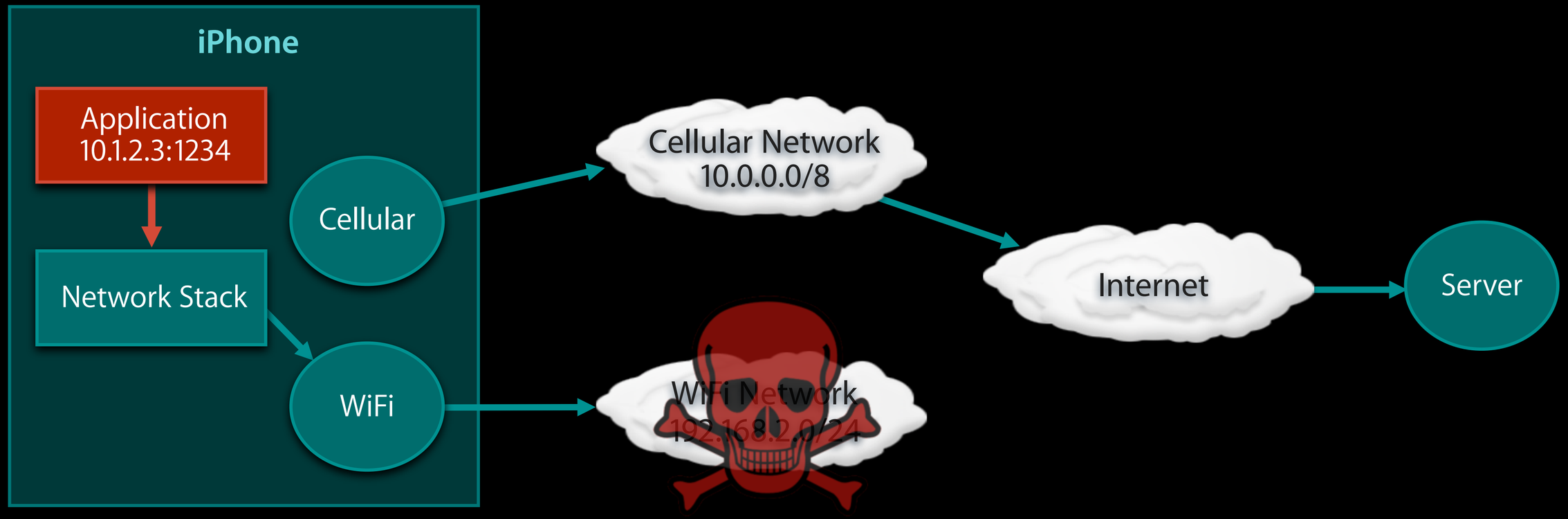
How It Works Today

Socket bound to cellular IP



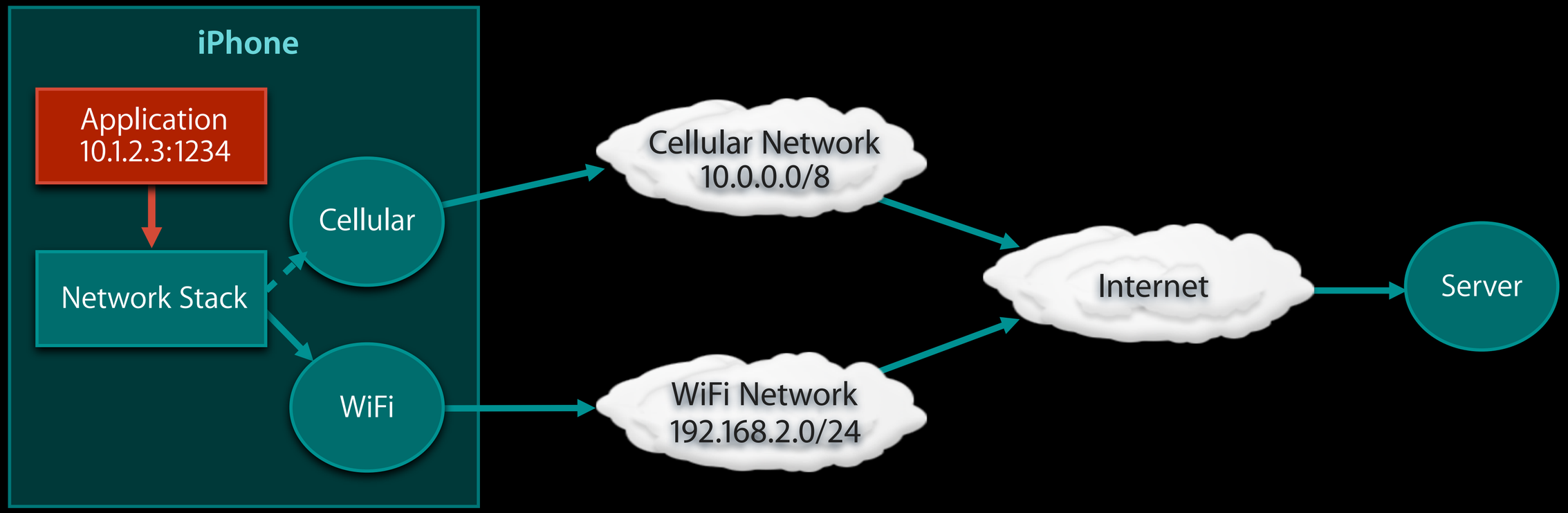
How It Works Today

Default route changes to WiFi



iPhone OS 3.0 Scoped Routing

Existing connections go over cellular



Coping with Routing

- Monitor reachability
- Reconnect on changes
- Scoped routing covers up mistakes
  - Mistakes result in slower connections

Reachability

- Networking hard to get right
- Not a pre-flight check
- Connect first
- Reachability second
  - Change notification
  - Avoid polling
  - React faster

Reachability

- Create SCNetworkReachabilityRef
  - Watch for remote reachability

    `SCNetworkReachabilityCreateWithName`
  - Watch reachability for established connection

    `SCNetworkReachabilityCreateWithAddressPair`
- Use asynchronous API

  `SCNetworkReachabilitySetCallback`

  `SCNetworkReachabilityScheduleWithRunLoop`

  `SCNetworkReachabilityGetFlags`
- Evaluate options on reachability callback

Reachability Flags—Cheat Sheet

`kSCNetworkReachabilityFlagsReachable`

- Safe to attempt connection
- No guarantee connection will succeed

`kSCNetworkReachabilityFlagsConnectionRequired`

- Interface not currently connected
- CoreFoundation/Cocoa APIs will trigger “dial”

`kSCNetworkReachabilityFlagsIsWWAN`

- Over cellular
- Over VPN over cellular

UIRequiresPersistentWiFi

- Info.plist key
  - True means “Network Application”
- Disables WiFi disassociation timer
- Allows WiFi UI
  - Password dialog
  - Ask to join dialog
  - Captive web sheet

Bluetooth Device to Device

- Bonjour over Bluetooth discovery
  - Advertise, browse, resolve
  - Expensive
    - Moderately—browse
    - Extremely—resolve
- IP over Bluetooth PAN (Personal Area Network)
  - Idle disconnect
  - One outgoing PAN connection
  - Maximum three PAN connections
- WiFi/Bluetooth coexistence

Performance

- Simulator
- Latency
- Non-blocking IO
- Multiple threads
- CFSocket

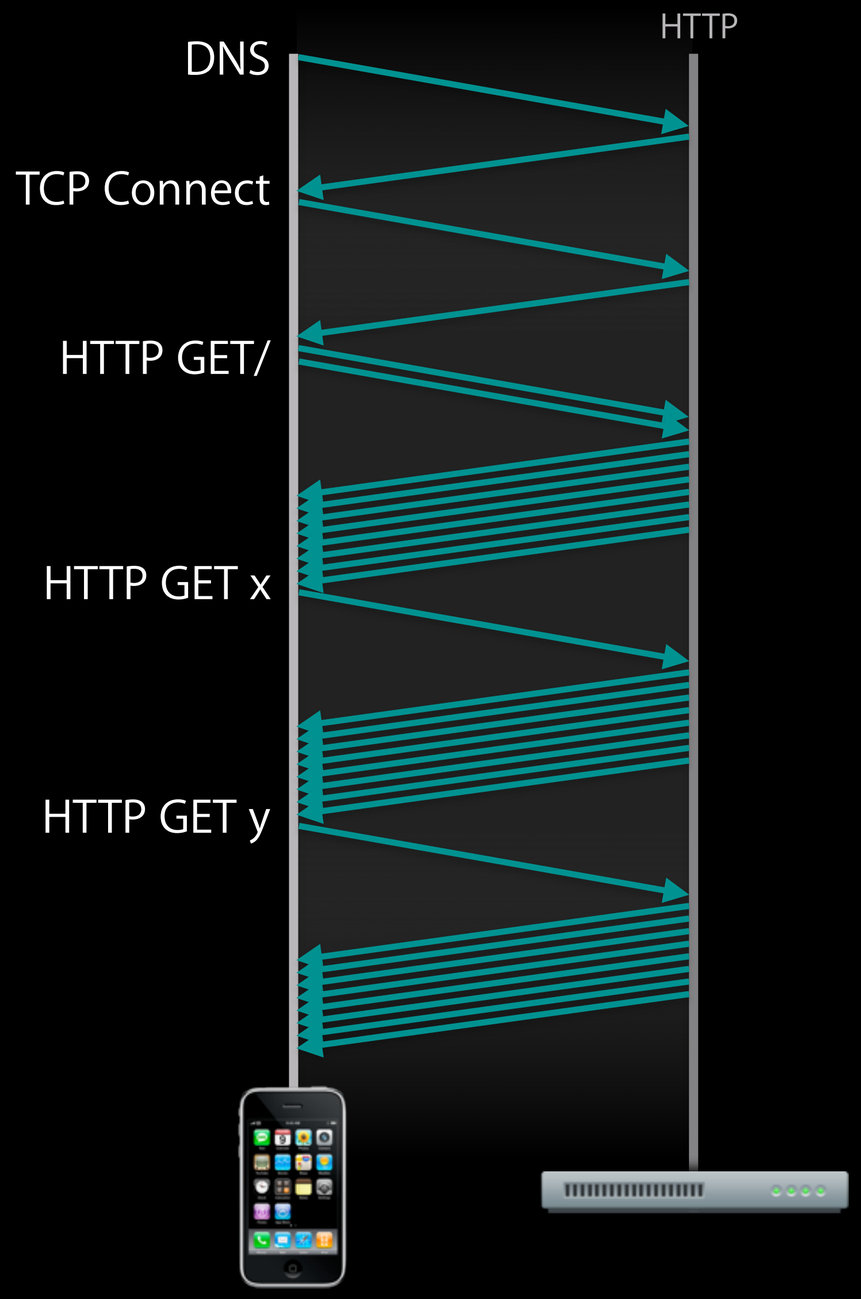
Simulator

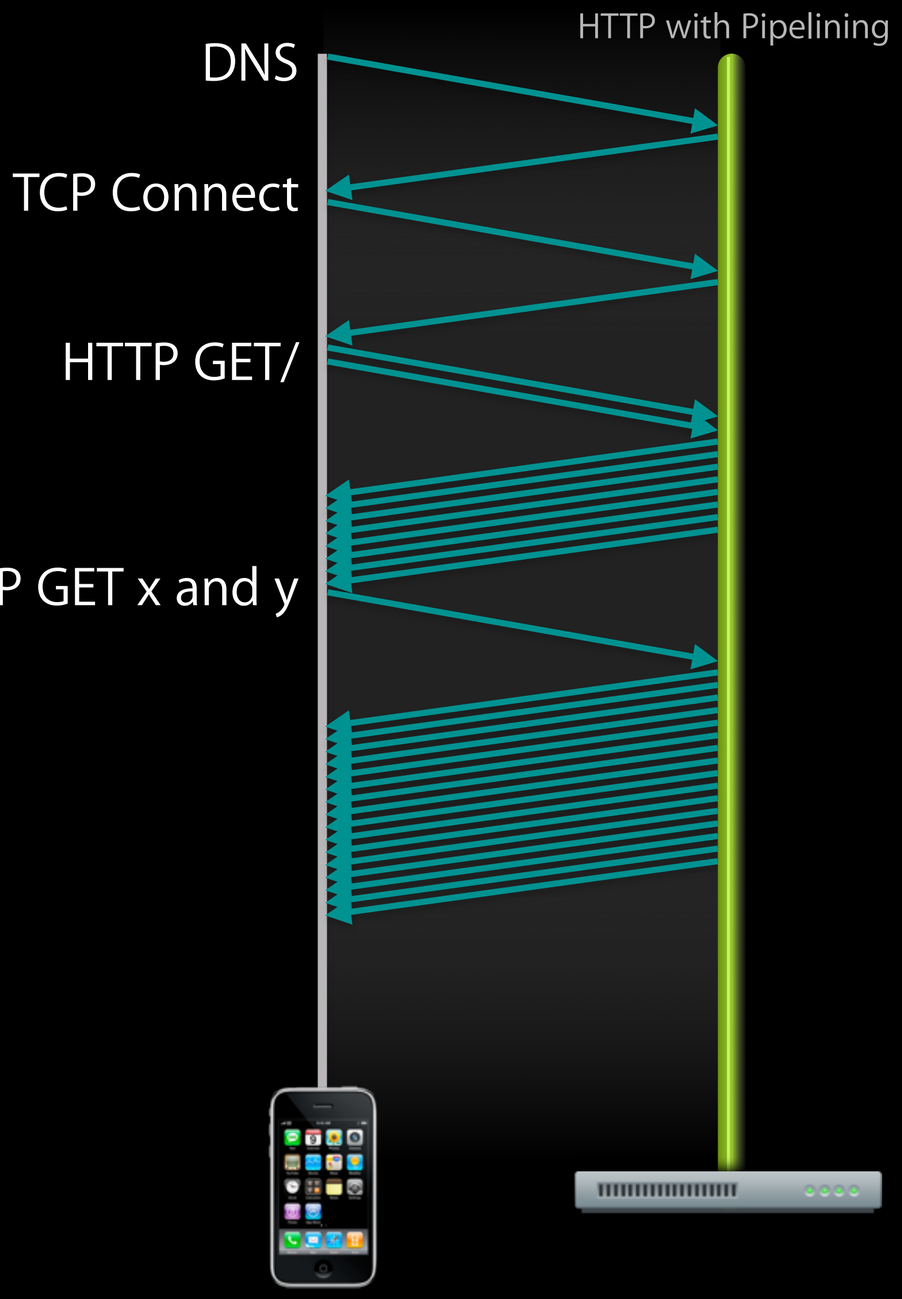
- Simulator for rapid application development
- Contains no network simulator
- Test, test, and test again on iPhone
  - WiFi
  - 3G/EDGE
  - VPN
- Differences
  - CPU
  - Memory and network buffers
  - Latency and bandwidth

Latency

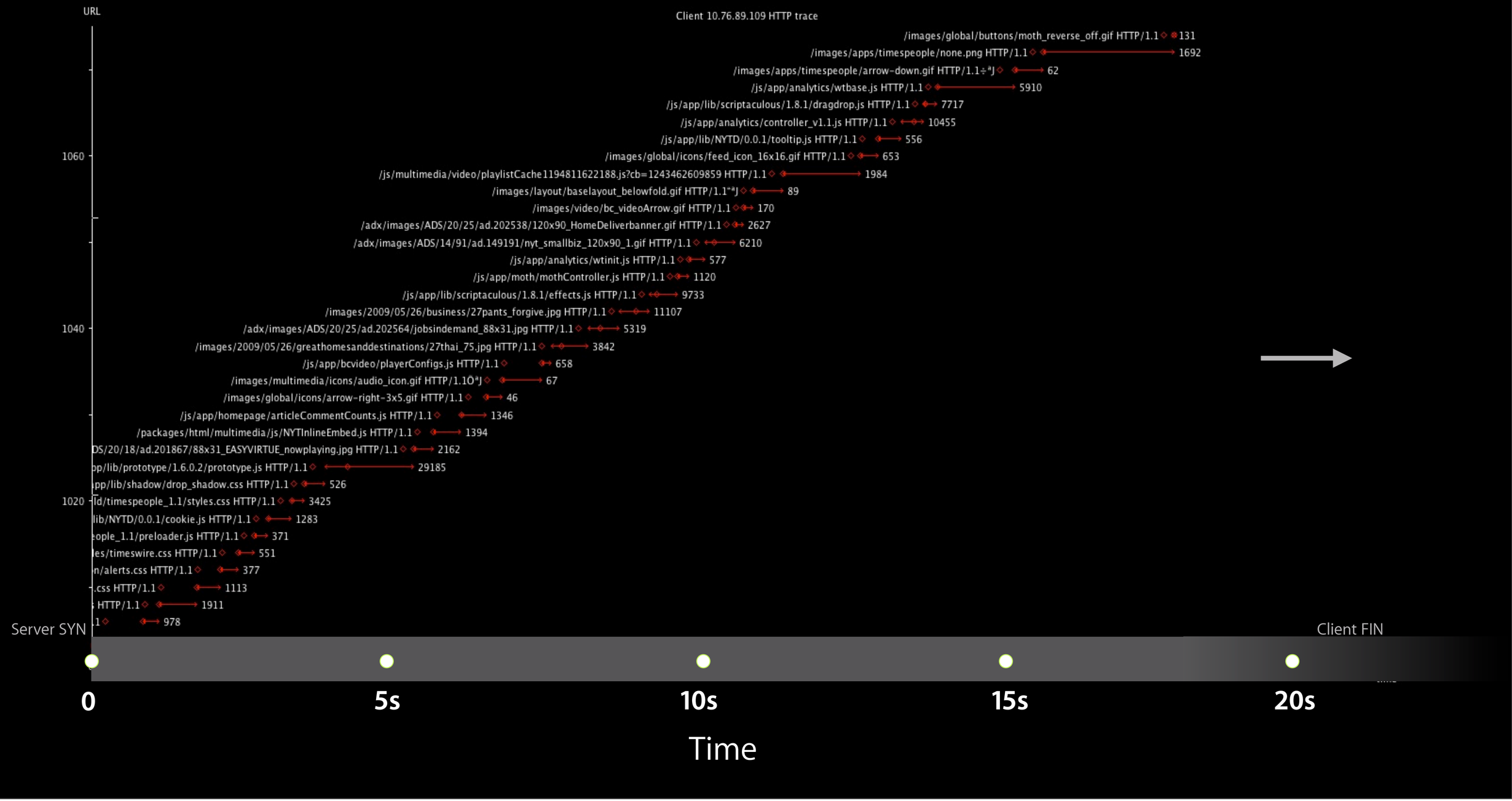
- Cellular round trip time is long
  - ~ 1/3 second EDGE
  - ~ 1/6 second 3G
- Latency kills
  - DNS (1/3 second)
  - TCP three-way handshake (1/3 second)
  - Request (1/3 second)
- HTTP is latency bound
  - Safari over 3G about 2x faster than EDGE

Latency Diagram

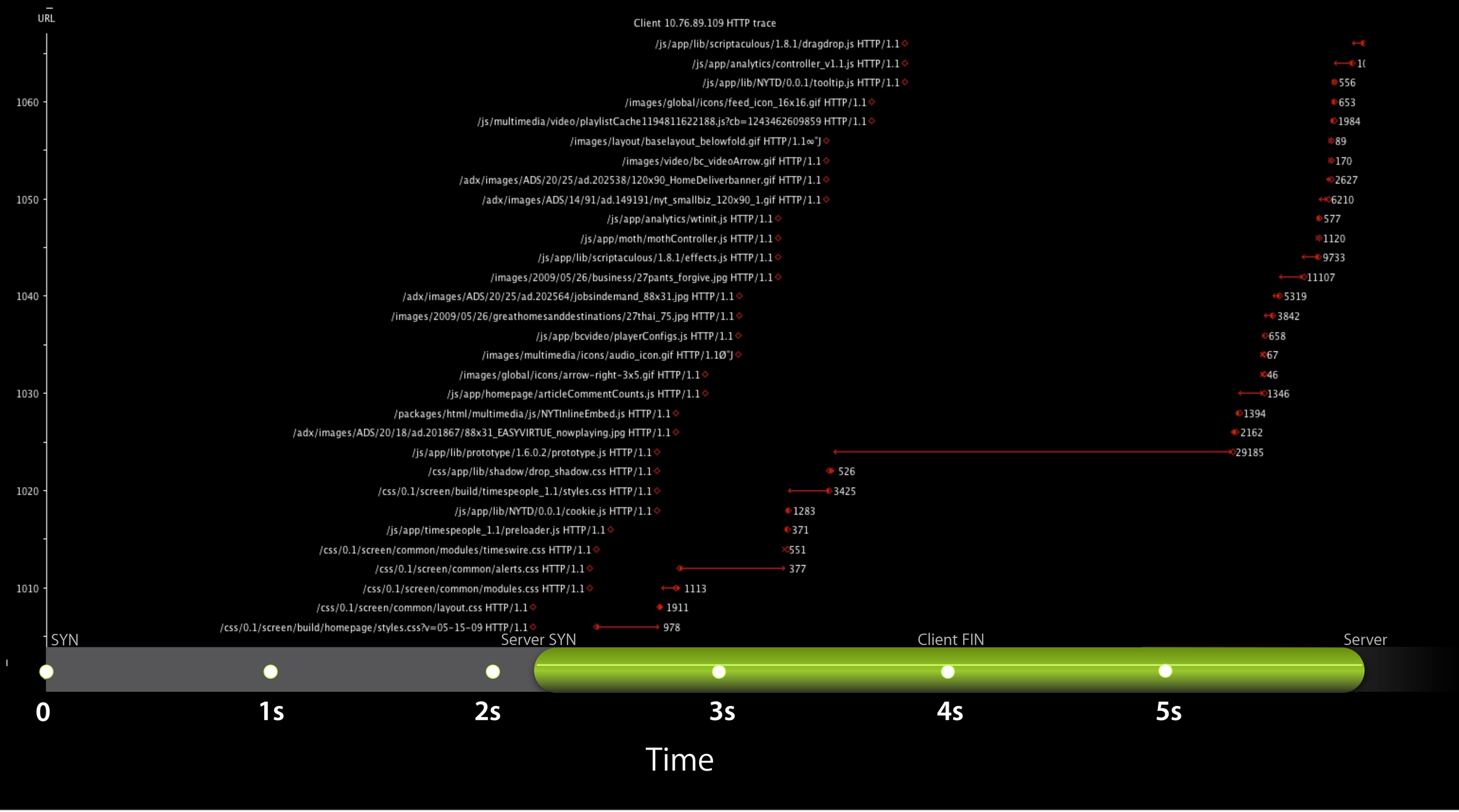




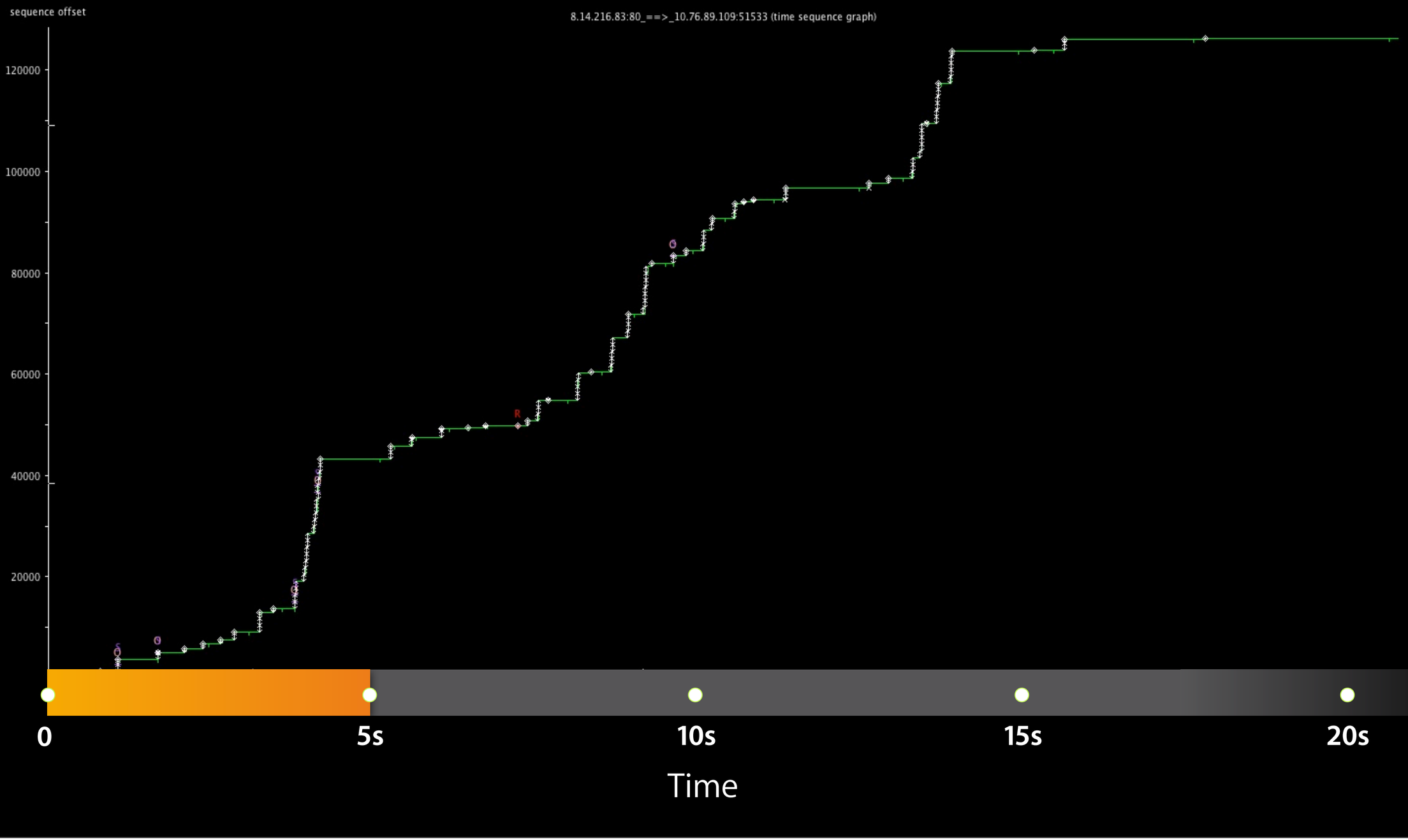
HTTP GET—NYTimes



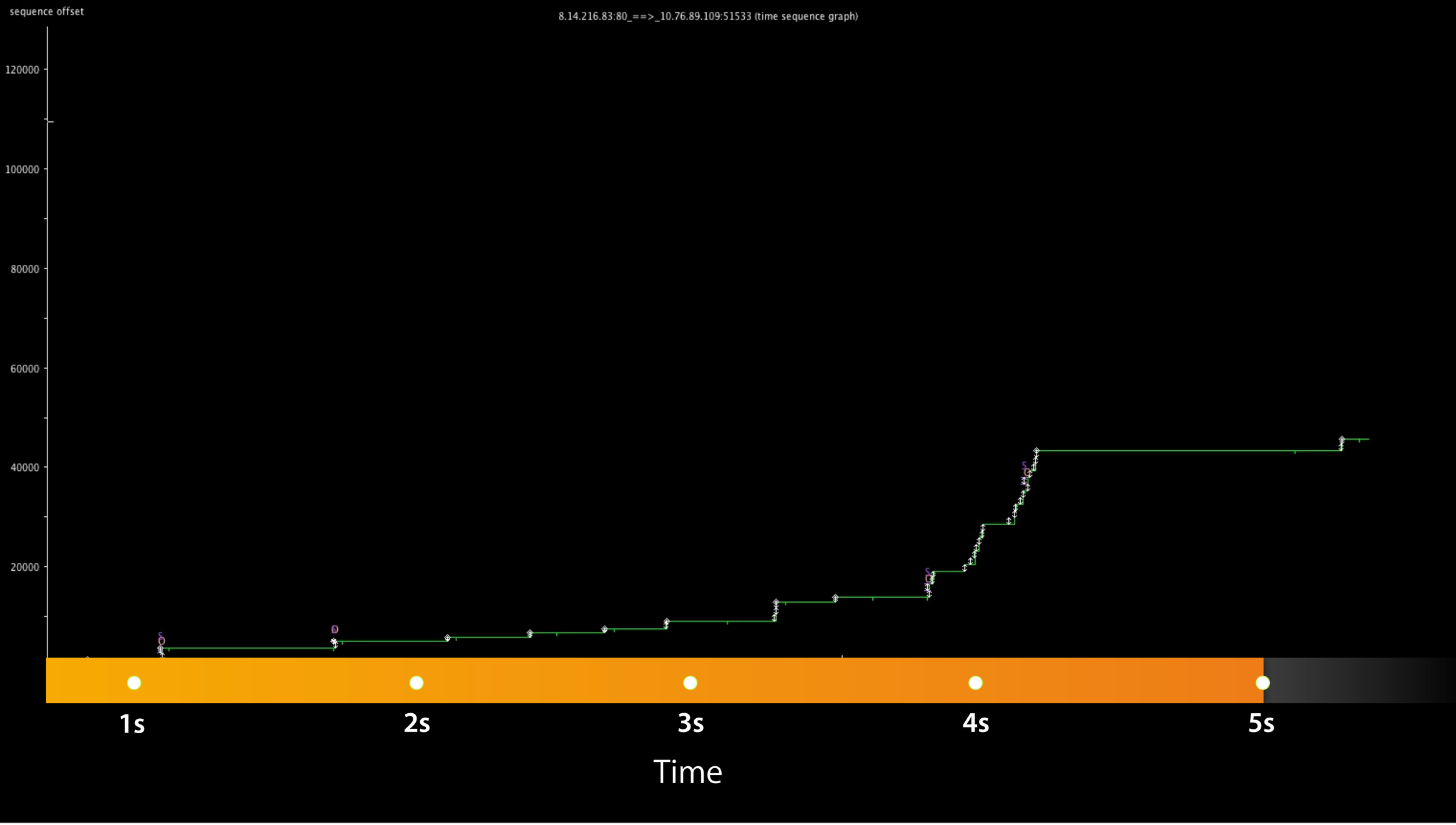
HTTP Get NYTimes Pipelined



HTTP Receive



HTTP Receive—First Five Seconds



HTTP Pipelined Receive

Latency Notes

- Protocol design
  - Allow multiple requests
  - Bad: Request A, wait, get response, request B
  - Good: Request A, request B, request C, request D
    - HTTP Pipelining busted
      - Proxies, Apache plug-Ins
- Multiple connections bad
  - Congestion window/slow start problem
  - Hard to tune
- Investigate—don't settle for slow

Non-Blocking IO

- Never block main thread
  - DNS—30 seconds
  - Blocking socket read/write/connect
- Indeterminate call duration
- Watchdog—20 seconds
- Avoid modal “connecting” dialogs
- Remain responsive during network failure

Threads vs. Asynchronous State Machine

- Threads
  - Takes more memory
  - Thread synchronization overhead
  - Easier to write
  - Better for CPU bound workloads
- Asynchronous state machine
  - Minimal resources
  - Better scalability
  - Better for I/O bound workloads
- iPhone/iPod Touch have one core

CFSocket Performance

- Run loop integration cost
  - Run loop mach port based
  - Sockets file descriptor based
- CFSocket thread
  - Watches file descriptors
  - Signal via mach port
- Use non-blocking socket
- Read and write until EWOULDBLOCK
- Not applicable to CFStream
- Better on SnowLeopard

Miscellaneous Tips

- WiFi vs. 3G vs. EDGE
  - Don’t assume performance advantage
- WiFi powers off
  - 30 minute timer
    - UIRequiresPersistentWiFi for network apps
  - Screen lock timer
- Cellular and VPN “Dial” triggers
  - Core Foundation
  - Cocoa
- Connect UDP sockets

Summary

- Use reachability
- Non-blocking IO
- Be aware of latency
- Test on iPhone hardware
- Assume dynamic network environment
- Connect UDP socket
- Read all data when using CFSocket

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09